**Chapter 3**

**Analysis**

* 1. **Requirements Analysis**

**3.1.1 Usability:**

The user of the system will find this system to be very easy to use since they just have to enter the keyword of their interest and the result will be shown to them in a matter of seconds.

**3.1.2 Reliability:**

This system is reliable because of the use of up to the mark NLP and Machine learning techniques. According to various researches, on applying appropriate NLP techniques the accuracy of Naive Bayes and Maximum entropy can cross 80%.

**3.1.3 Performance:**

The performance of this system will be fairly good in terms of speed. The only thing that will make it slow is the restriction of twitter to fetch the tweets.

**3.1.4 Implementation:**

Implementation is done by using python 2.7, NLP and Machine Learning approach. Machine Learning tasks will be handled by NLTK 3 and libsvm libraries. All these libraries have efficient classes which will carry out the tasks in minimum time. Also for interface we will use webpy, which is again a python library for plotting graphs of the data analyzed.

**3.1.5 Interface:**

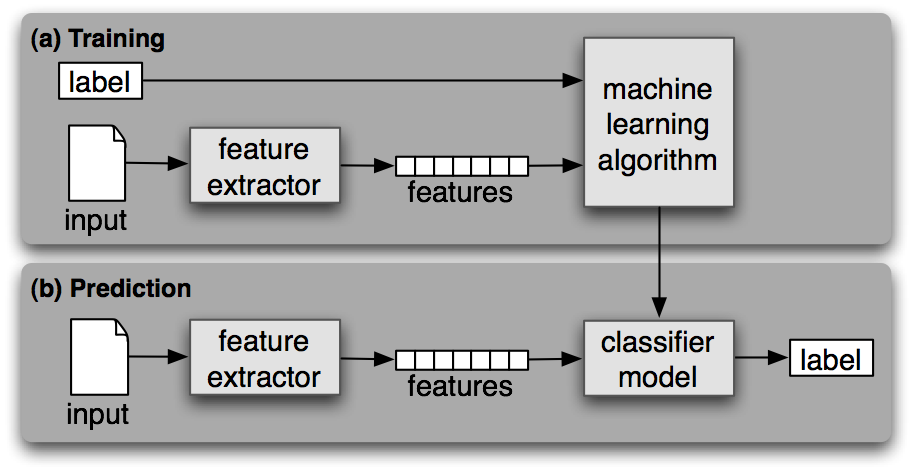
The interface will consist of a text box which will take the input (i.e. a keyword) from the user and a button to submit that keyword. The Output will be shown in terms of graph which shows the changes in sentiments over the past few days or according to how we program it.

**3.2 Proposed System**

A Sentiment Analyzer is what we need to classify sentiments. This system will contain a trained machine learning classifier. When a user enters particular keyword the system will retrieve the data about it from sources like twitter and the trained classifier will calculate the result about the queried keyword.

Our approach is to use different machine learning classifiers and feature extractors. The machine learning classifiers are Naive Bayes and Maximum Entropy. All of these classifiers require training data and hence these methods fall under the category of supervised classification.

The feature extractors are unigrams and unigrams with weighted positive and negative keywords. We build a framework that treats classifiers and feature extractors as two distinct components. This framework allows us to easily try out different combinations of classifiers and feature extractors.

****

**Fig. 3.1 Supervised Classification [9]**

We can compare the accuracy of different classifiers. The accuracy also depends on the quality and amount of training dataset.